Learning Cloud Dynamics to Optimize Spot Instance Bidding Strategies

Mikhail Khodak*, Liang Zheng*, Andrew S. Lan*, Carlee Joe-Wong†, and Mung Chiang‡
*Princeton University, †Carnegie Mellon University, ‡Purdue University
*{mkhodak,liangz,andrew.lan}@princeton.edu, †cjoewong@andrew.cmu.edu, ‡chiang@purdue.edu

***Abstract*—As infrastructure-as-a-service clouds become more popular, cloud providers face the complicated problem of maximizing their resource utilization by handling the dynamics of user demand. Auction-based pricing, such as Amazon EC2 spot pricing, provides an option for users to use idle resources at highly reduced yet dynamic prices; under such a pricing scheme, users place bids for cloud resources, and the provider chooses a threshold "spot" price above which bids are admitted. In this paper, we propose a nonlinear dynamical system model for the time-evolution of the spot price as a function of latent states that characterize user demand in the spot and on-demand markets. This model enables us to adaptively predict future spot prices given past spot price observations, allowing us to derive user bidding strategies for heterogeneous cloud resources that minimize the cost to complete a job with negligible probability of interruption. Along the way, the model also yields novel, empirically verifiable insights into cloud provider behavior. We experimentally validate our model and bidding strategy on two months of Amazon EC2 spot price data and find that our proposed bidding strategy is up to 4 times closer to the optimal strategy in hindsight compared to a baseline regression approach while incurring the same negligible probability of interruption.**

I. Introduction

As businesses increasingly utilize cloud infrastructure to run their applications, cloud providers (CPs) have begun to offer them a variety of cloud-based services [1]–[3]. Infrastructure-as-a-service (IaaS), as its name suggests, provides users with time on virtualized compute resources that are hosted on the CP's physical servers. Since users can freely choose the amount of compute resources that they rent from the CP, IaaS can provide them with *on-demand resource elasticity*, i.e., virtualization of computing resources that can expand or contract to meet real-time user demands. For instance, a business running data analytics on its webpage browsing patterns might decide to rent more compute resources at times when many users are accessing its website.

However, this resource elasticity comes with a significant challenge as IaaS becomes more popular: *how can the CP simultaneously meet the resource needs of many jobs, each of which may experience large, possibly unpredictable fluctuations in their resource needs?* Many works on cloud operations have attempted to address this issue, e.g., through job scheduling [4], [5] or optimized virtual machine placement [6], [7]. In this work, we propose another solution: using pricing policies to shape resource demands.

A. Cloud Pricing and Resource Utilization

CPs today offer a variety of different IaaS pricing policies that can incentivize different types of user behavior. For instance, volume-discounted pricing charges users at a reduced unit price per instance if their usage exceeds a given threshold; such a pricing policy then incentivizes users to submit larger, longer-lived jobs that can take advantage of the volume discount. Examples of such pricing include Google Cloud Platform's sustained-use discount [1] and Amazon EC2's reserved instances [2]. However, while CPs can help to stabilize their resource demands with volume discounts, they cannot entirely avoid fluctuations in users' resource needs. Since datacenters have a fixed physical capacity at an hour-to-hour or day-to-day timescale, the CP must provision their infrastructure to users' peak demands, leaving some idle resources at off-peak times.

To improve off-peak utilization, CPs can offer their users reduced prices at these times, e.g., Google's preemptible virtual machines' [8] and Amazon EC2 spot instances' auction-based spot pricing [3]. Since it would be difficult for the CP to predict its amount of idle resources at any given time, neither of these schemes guarantees their users access to cloud resources; instead, they promise to provide resources to users if they are available in the future. Users then accept a lower price in exchange for unpredictable resource availability. Spot instance auctions in particular, however, allow users to optimize their bids so as to reduce the amount of job interruptions due to resources being unavailable. Indeed, by modeling spot dynamics, we can derive insights into not only spot instance markets but the operations of the cloud itself.

B. Our Contributions

In the cloud spot market, the CP dynamically sets "spot" prices according to changes in supply, i.e., available spare resources in the on-demand market, and demand, i.e., bid arrivals and departures. Users submit bids at their chosen bid prices; their jobs start running if their bid prices exceed the current spot price, until the spot price exceeds their bid prices. When that occurs, their jobs are interrupted and resume once their bid prices again exceed the given spot price. Users are always charged at the current spot price when their jobs are running. By predicting the future spot prices, they can predict

The work in this paper was in part supported by DARPA DCOMP Program under Contract No. HR001117C0052 and No. HR001117C0048, NSF grants CCF-1302518 and CCF-1527371, and ONR-N00014-16-1-2329.

the number of times their jobs will be interrupted, using this knowledge to optimize their bids.

Predicting future spot price dynamics, however, is difficult, as the value of the spot price is impacted by many factors. Generally, more demand for on-demand resources means there are fewer spare resources available for the spot market, resulting in higher spot prices. Job arrivals and departures can also impact the spot prices. For example, a large number of new bids in one time slot can significantly affect the spot price at the next time slot; since user bids are reconsidered in every time slot until their associated jobs complete, current bids can affect the future spot prices, introducing temporal correlations between the spot prices at different times.

It is unclear how the CP uses these factors to determine the spot prices. On the one hand, the CP may set high spot prices to increase their per-job profit at the expense of idle cloud resources; on the other hand, the CP may set low, "clearance" spot prices to fully utilize their available resources. CPs in general do not reveal their pricing strategies publicly, but Amazon EC2 does allow users to view its spot price history for the past two months. In this paper, we use these historical traces to answer two key questions: first, *how should we model demand dynamics in both the on-demand market and the spot market so as to predict future spot prices?* Second, given this predictive ability, *how should a user bid in order to complete a job at minimal cost?* In answering these questions, we make the following research contributions:

Prediction of spot prices (Section III): We first prove under very general assumptions that *the statistics of Amazon EC2's spot price history do not correspond to a CP profit-maximizing approach for setting spot prices*. This confirms observations made in [9] and justifies an assumption that the CP sets spot prices to fully utilize cloud capacity. We proceed to *develop a hidden Markov model that views the spot price as a stochastic function of latent states* corresponding to user demand in both the on-demand and spot models, which themselves evolve stochastically as cloud users arrive and depart the bidding queue. This model is posed as a nonlinear dynamical system parameterized by variables associated with the supply and demand for cloud resources. Thus, unlike prior works on bidding optimization [10], our model not only explicitly accounts for the interplay between the on-demand and spot markets under a resource capacity constraint but also handles time correlations between spot prices.

Optimal bidding strategies (Section IV): To lowers users' costs of using spot instances while maintaining a low probability of job interruption, we next consider bidding among heterogeneous machine types. Amazon EC2 offers general purpose `m3.*` instances, compute optimized `c3.*` instances, memory optimized `r3.*` instances, and storage optimized `i3.*` instances, with different configurations for different resources. Given a family of instance types (e.g., `c3.*`), *we use our spot price prediction model to optimally choose the type of instance to bid for* in this family. For example, `c3.2xlarge` has double the vCPU of `c3.xlarge` and thus can reduce a CPU-constrained job's runtime. Although the spot price for a higher-configuration instance like `c3.2xlarge` is usually higher than for a lower-configuration instance, the shorter job runtime on the higher-configuration instance combined with the price fluctuations captured by our model may make it more cost-effective to bid for this instance. Additionally, jobs may experience less variance in spot prices across a shorter runtime.

To experimentally verify both of these results, we download the Amazon EC2 spot price history from 16 February to 26 July, 2017. We divide this dataset into training and testing sets of equal size, with the former comprising the data up to 7 May. The test set is used to evaluate our optimal bidding strategies using a spot price model trained on the training set, and we show that it fits the empirical spot prices well. We also compare the bidding results to an auto regressive baseline [9], showing that bidding strategies made with our prediction method outperforms, especially for volatile instances. We conclude the paper in Section V. All proofs and technical details can be found in the technical report [11].

II. Related Work

Since the advent of Amazon EC2 spot pricing in 2009, many works [9], [12]–[14] have characterized spot prices from a statistical perspective. These approaches, however, have limited predictive power, as they do not consider the dynamics between the supply and demand of cloud instances. As we discuss above, spot pricing reflects fluctuations in cloud resource utilization, requiring a model that considers these dynamics. Moreoever, the assumptions of uncorrelated Gaussian noise needed by auto-regressive models, such as used by [9], do not hold on 2017 spot data (see Figure 3).

The authors of [10] used a profit-maximization model to understand spot price distributions; however, they considered only the asymptotic (time-invariant) user demand regime and made fairly restrictive assumptions on spot user arrival and bid distributions. Moreover, there has been evidence showing that the CP may not use the profit-maximizing spot prices [9]. Since a purely statistical understanding of spot prices may not capture their true origins, we model spot prices as a function of supply and demand by characterizing the dynamics of both the on-demand and spot markets.

A few works have argued that the cost benefits brought by spot offerings can be realized with intuitive bidding strategies [15]. However, choosing between spot instances and bid levels affects both the cost and interruptibility of users' jobs and may allow users to realize significant cost savings, making the bidding problem non-trivial. Other works have derived more in-depth bidding strategies for resources in the cloud spot market by explicitly considering job deadlines [16], cost minimization [10], [17], and task dependency [18], [19]. Some works have attempted to bid across different geographical regions and instance types [20]. To limit the variance in the allocated resources, [21], [22] suggest that users rent resources in both the on-demand market and the spot market. However, the spot price models used in these works are fairly simplistic and achieve limited predictive power, limiting the performance achieved by their proposed bidding strategies. Our work differs

TABLE I
KEY TERMS AND SYMBOLS

Symbol	Definition
N	The number of instances of the same VM type.
$\overline{\pi}$, $\underline{\pi}$	On-demand price and instance operation cost ($\underline{\pi} < \overline{\pi}$).
π_t	Spot price at time slot t ($\pi_t \geq \underline{\pi}$)
$N_t^{(d)}$, $N_t^{(s)}$	Number of running on-demand and spot instances.
$\mathcal{B}_t \subset \mathbb{R}_+$	Set of spot bids, with cardinality $B_t = \lvert\mathcal{B}_t\rvert$.
$\Lambda_t^{(d)}$, $\tilde{\Lambda}_t^{(d)}$	Number of request arrivals and job departures in the on-demand market over time interval t; assumed to follow independent time-invariant Poisson distributions with parameters θ_d and $\tilde{\theta}_d$.
$\Lambda_t^{(s)}$, $\tilde{\Lambda}_t^{(s)}$	Number of bid arrivals and job departures in the spot market over time interval t. Assumed to follow independent time-invariant Poisson distributions with parameters θ_s and $\tilde{\theta}_s$.

from these past efforts by first learning a spot price model that is rooted in understanding CPs' decision criteria, and then devising a bidding strategy that considers cost minimization in the more natural setting of heterogeneous cloud resources.

III. A Dynamical Model for Spot Prices

In this section, we develop a dynamical model for the way in which a CP sets spot prices by considering both its cloud computing resources and user demand in the on-demand and spot markets. These cloud resources are sold in the unit of instances, which are remote virtual machines (VMs) with specified amounts of CPU, memory, storage, etc. We suppose that there are in total N instances of the same VM type that are offered to all users requesting it over discrete time slots $t \in \{0, 1, \ldots\}$. Each instance is assumed to have operational cost $\underline{\pi}$ incurred by the CP per unit per hour. At each time slot t, user requests for on-demand instances are satisfied instantaneously and charged at the on-demand price $\overline{\pi} > \underline{\pi}$, while user requests for spot instances at their bid prices are allocated to spare resources. Bids with bid prices higher than the current spot price $\pi_t \geq \underline{\pi}$ are regarded as successful and can start running. We summarize our notation in Table I.

When setting the spot price, the CP needs to take into account both the $N_t^{(d)}$ currently running on-demand instances and a set $\mathcal{B}_t \subset \mathbb{R}_+$ of bids from $B_t = \lvert\mathcal{B}_t\rvert$ spot instance requests. Many works [10], [23], [24] assume that the CP's objective is to maximize its own profit under some capacity constraint, which we show is inconsistent with empirical spot price distributions in Section III-A. Since unsuccessful bids are pushed into a queue and considered together with new bids in the next time slot, there will be auto-correlation for spot instance demand across time. By formalizing the relationship between user demands and spot prices in Section III-B, we characterize this time correlation and its effect on spot prices via a dynamical system. We then describe an approach to solving the dual problem of estimating both the hidden states and the parameters characterizing request arrival and job completion rates via the expectation-maximization (EM) algorithm in Section III-C. Learning this *cloud dynamics model* will facilitate our bidding strategy design in Section IV.

A. Provider Profit Optimization

We first construct a model for how the CP sets the price. At a given time slot t, the CP determines the spot price π_t given the on-demand usage $N_t^{(d)}$ and the set of bids $\mathcal{B}_t$. It then receives profit

$$(\overline{\pi} - \underline{\pi})\, N_t^{(d)} + (\pi_t - \underline{\pi})\, N_t^{(s)}, \tag{1}$$

where we define the spot market usage as

$$N_t^{(s)} = \lvert\{b \in \mathcal{B}_t : b \geq \pi_t\}\rvert\,, \tag{2}$$

i.e., the number of successful bids at time t. Since $\underline{\pi}$ and $\overline{\pi}$ are fixed (as long as the timescale is not too large) and $N_t^{(d)}$ is determined exogenously by user demand, the CP's spot price decision affects only the second term, which is thus the objective of the maximization problem.

We now need to determine the constraint on this maximization problem. Since for certain $\mathcal{B}_t$ the CP can potentially make more profit selling fewer instances at a higher spot price, the *profit-maximizing constraint* is $N_t^{(d)} + N_t^{(s)} \leq N$. However, the CP may wish to sell all of its excess capacity, e.g., to have a larger market share or for other economic considerations, even if this means a lower profit. Assuming $N_t^{(d)} + B_t \geq N$ the corresponding *full capacity constraint* is $N_t^{(d)} + N_t^{(s)} = N$. Although both settings are economically interesting, we next show that under some reasonable assumptions, Amazon EC2 spot price data indicates that Amazon is not using the profit-maximizing constraint with high probability (in B_t), justifying our use of the full capacity constraint instead.

To motivate this decision, we first make the assumption that each spot bid $b \in \mathcal{B}_t$ is drawn independently from a (possibly time-varying) distribution that *weakly stochastically dominates* the uniform distribution over the interval $[\underline{\pi}, \overline{\pi}]$ (henceforth denoted $\mathcal{U}[\underline{\pi}, \overline{\pi}]$), where we employ the following definition:

Definition 1: The distribution of a real-valued random variable X with cumulative density function F_X is said to *weakly stochastically dominate* (WSD) the distribution of a real-valued random variable Y with cumulative density function F_Y if $F_X(x) \leq F_Y(x)\ \forall\ x \in \mathbb{R}$.

WSD implies that for any $x \in \mathbb{R}$, a random variable drawn from the dominating distribution is more likely to take a value above x than one drawn from the dominated distribution. Distributions with support in $[\underline{\pi}, \overline{\pi}]$ that dominate $\mathcal{U}[\underline{\pi}, \overline{\pi}]$ are more concentrated on the upper half of the interval (in general any distribution whose density function is supported and non-decreasing on $[\underline{\pi}, \overline{\pi}]$, including $\mathcal{U}[\underline{\pi}, \overline{\pi}]$ itself, satisfies our assumption).

Our use of this weak assumption has basis in both previous analyses of spot markets and other auctions (e.g. [10] assumes bids are drawn from $\mathcal{U}[\underline{\pi}, \overline{\pi}]$) as well in the simple bidding strategy of bidding the on-demand price since the user only pays the spot price anyway, as is commonly advocated (e.g. in [15]) and used in practice. Under the given model, the WSD assumption yields the following proposition, which, informally, shows that if the CP is using a profit-maximizing constraint then the spot price is bounded away from $\underline{\pi}$ with high probability:

Proposition 1: Suppose at time slot t, all bids $b \in \mathcal{B}_t$ are drawn independently from a distribution that weakly stochastically dominates $\mathcal{U}[\underline{\pi}, \overline{\pi}]$. Then, defining $\pi^{(\rho)} = \rho\overline{\pi} + (1-\rho)\underline{\pi}$ for $\rho \in [0, 1]$, if the CP sets the spot price π_t so as to maximize (1) subject to the profit-maximizing constraint $N_t^{(d)} + N_t^{(s)} \le N$, then for any $\rho \le \frac{1}{4}$ we have that

$$\mathbb{P}\left(\pi_t \le \pi^{(\rho)}\right) \le \exp\left(-2\left(\frac{1}{2} - 2\rho\right)^2 B_t\right). \qquad (3)$$

Under the WSD assumption this proposition gives us the following explicit condition that must be satisfied in order to assume that a CP operating under our model sets a profit-maximizing constraint: given some ρ, if there are always at least B spot users for each instance, then the proportion of all time slots at which we have $\pi_t \le \pi^{(\rho)}$ must be less than $\exp\left(-2\left(\frac{1}{2} - 2\rho\right)^2 B\right)$. We start by examining the spot price history of the general purpose computing instance `m3.medium` in the US-East-1 region (zone a). We find that the condition is violated even if $B = 1$: choosing $\underline{\pi}$ to be the minimum price, for $\rho = 0.15$, we find that $\pi_t \le \pi^{(\rho)}$ around 96% of the time (Figure 1) while the condition predicts this event to occur at most 92% of the time.

More generally, letting $B = 10$ we find that 81% of the spot price histories of all active instance types in the US-East-1 region (zone a) violate the condition (Figure 1). Since the bound in Proposition 1 is not tight and since the number of spot users B may be much larger than 10, this is strong empirical evidence that Amazon EC2 spot prices are not set using a profit-maximizing constraint; instead, we postulate that they are using the spot market only to use up excess capacity, i.e. they set the full capacity-utilizing constraint. Such a strategy may be driven by longer term profit maximization, e.g. if the CP expects that risk-averse users will not use the spot market if the spot price is often very close to the on-demand price, or by the desire for a larger market share.

Following these observations and assuming $B_t \ge N - N_t^{(d)}$ (i.e. there is always enough spot-demand to fill capacity), we formulate the CP profit optimization problem as:

$$\begin{aligned} \underset{\pi_t}{\text{maximize}} \quad & (\pi_t - \underline{\pi})N_t^{(s)} \\ \text{subject to} \quad & N_t^{(d)} + N_t^{(s)} = N. \end{aligned} \qquad (4)$$

Assuming the CP knows $\mathcal{B}_t$, the solution to this optimization problem is given by

$$\pi_t = \max_{|\{b \in \mathcal{B}_t : b \ge \pi\}| = N - N_t^{(d)}} \pi. \qquad (5)$$

Our goal is to model the evolution of the spot price as an observed variable depending on hidden stochastic state variables $N_t^{(d)}$ and B_t. To simplify terminology we first normalize these variables by the total number of instances, i.e. define $n_t = N_t^{(d)}/N$ and $b_t = B_t/N$. Then, to reduce the dimensionality (in order to build a model where we infer b_t instead of $\mathcal{B}_t$), we follow [10] and assume that all bids $b \in \mathcal{B}_t$ are drawn independently from the uniform distribution on $[\underline{\pi}, \overline{\pi}]$. We can then view π_t as a noisy observation of a function of n_t and b_t, as stated in the following Lemma:

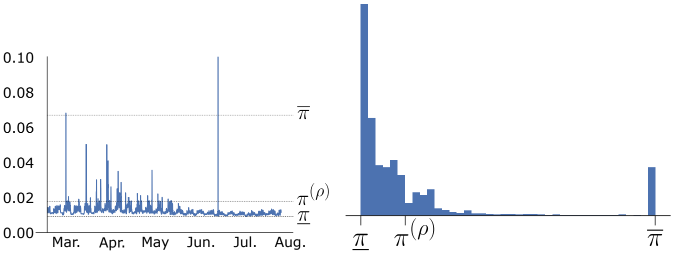


Fig. 1. Left: Time evolution of `m3.medium` spot price over 5 months in 2017. Note how the spot price π_t is usually less than $\pi^{(\rho)}$ for $\rho = 0.15$. Right: Histogram of spot prices gathered from all instance types (normalized to lie within the same price range) from the US-East-1 region (zone a). Most of the mass is below $\pi^{(\rho)}$ for $\rho = 0.15$.

Lemma 1: Fixing $\frac{N - N_t^{(d)}}{B_t} = C_t$ for some constant $C_t \le 1$, as $B_t \to \infty$ the distribution of π_t conditioned on n_t and b_t converges in distribution to $\mathcal{N}\left(\overline{\pi} - (\overline{\pi} - \underline{\pi})\frac{1-n_t}{b_t}, \frac{(\overline{\pi}-\underline{\pi})^2}{4Nb_t}\right)$.

Thus, assuming there are always enough spot bids to fill up excess capacity, this lemma implies that for large enough B_t the spot price π_t will be approximately normally distributed with mean $\overline{\pi} - (\overline{\pi} - \underline{\pi})\frac{1-n_t}{b_t}$. However, since this is an asymptotic limit and in practice there may not be enough spot bids to cover unused instances (i.e. $B_t < N - N_t^{(d)} \iff \frac{1-n_t}{b_t} > 1$), we use Lemma 1 to inform the construction of the following conditional model for the observed spot price:

$$\pi_t = \begin{cases} \underline{\pi} & n_t + b_t \le 1 \\ \overline{\pi} - (\overline{\pi} - \underline{\pi})\frac{1-n_t}{b_t} + \varepsilon_t & 0 < 1 - n_t < b_t \\ \overline{\pi} & n_t \ge 1 \end{cases} \qquad (6)$$

where $\varepsilon_t \sim \mathcal{N}\left(0, \frac{\sigma^2}{b_t}\right)$ is normally distributed with learnable variance parameter σ^2 corresponding to the noise in Lemma 1. Note that in this expression, the spot price rises to the on-demand price when the on-demand requests take up all the cloud capacity ($n_t = 1$) and is the minimum possible price when there is not enough spot demand to take up fill up excess capacity ($n_t + b_t \le 1$). We next discuss how to model the time evolution of n_t and b_t in order to understand the temporal dynamics of π_t.

B. Job Arrival and Departure

To capture the temporal evolution of the spot price, we now model the dynamics of cloud instance utilization. The dynamics consists of two parts: job arrival, which stems from user requests in the on-demand market and user bids in the spot market, and job departure, which occurs when jobs in either market are completed.

Over the course of one time slot t, a number, $\Lambda_t^{(d)}$, of users' jobs will arrive in the on-demand market to start running, while a number, $\tilde{\Lambda}_t^{(d)}$, of currently-running on-demand jobs will finish and exit the system. Thus, the number of running on-demand jobs evolves as $N_{t+1} = N_t + \Lambda_t^{(d)} - \tilde{\Lambda}_t^{(d)}$.

For spot instances, we model the dynamics of the number of active bids B_t, i.e., a queue of jobs competing with each other over spot instances via their bids. Over each time slot t, a number, $\Lambda_t^{(s)}$, of new users will submit bids, and a number, $\tilde{\Lambda}_t^{(s)}$, of existing bids will leave, either due to job completion or manual termination by users. Thus, the number of active bids evolves as $B_{t+1} = B_t + \Lambda_t^{(s)} - \tilde{\Lambda}_t^{(s)}$. We note that these B_t bids also include the jobs that are running on spot instances, since they will be interrupted by the system once their bid prices fall below future spot prices.

As before, we normalize all variables by the total amount of instances, i.e., define $\tilde{\lambda}_t^{(d)} = \tilde{\Lambda}_t^{(d)}/N$, $b_t = B_t/N$, $\lambda_t^{(s)} = \Lambda_t^{(s)}/N$, and $\tilde{\lambda}_t^{(s)} = \tilde{\Lambda}_t^{(s)}/N$, and use these notations throughout the rest of the paper. Hence, we arrive at the following dynamical system:

$$\begin{aligned} n_{t+1} &= n_t + \lambda_t^{(d)} - \tilde{\lambda}_t^{(d)} \\ b_{t+1} &= b_t + \lambda_t^{(s)} - \tilde{\lambda}_t^{(s)}. \end{aligned} \tag{7}$$

Since the unnormalized arrival-departure random variables are likely best captured by Poisson distributions [25]–[27], we find the closest counterparts for $\lambda_t^{(d)}, \tilde{\lambda}_t^{(d)}, \lambda_t^{(s)}$, and $\tilde{\lambda}_t^{(s)}$ and model them as independent time-invariant exponential random variables with parameters $\theta_d, \tilde{\theta}_d, \theta_s$, and $\tilde{\theta}_s$, respectively.

C. State and Parameter Estimation

Our proposed model on hidden state evolution (7) and spot price observation (6) defines a nonlinear dynamical system (NDS) parametrized by $\Theta = \{\theta_d, \tilde{\theta}_d, \theta_s, \tilde{\theta}_s, \sigma^2\}$. In this section, we provide a more detailed discussion of our model setup and describe how to jointly infer the hidden states and parameters of the model using EM.

First, we note that the spot prices of many types of cloud instances have a unique but fairly consistent lower bound, e.g. as shown in Figure 1. It is thus reasonable to regard the lowest spot price for each instance type as the operation cost $\underline{\pi}$ for that instance. More notably, there are times at which the spot price goes above the on-demand price; for example the spot price time series shown in Figure 1 experienced two such peaks, once in mid-March and once in late June. These events are likely caused by all cloud capacity being taken up by on-demand use, i.e. $n_t = 1$. At such times, the arrival-departure model is no longer valid, since no more on-demand instances can be started and no spot jobs can be completed. However, our model does handle the probability of arriving at such a state – it is $\mathbb{P}\left(\lambda_t^{(d)} - \tilde{\lambda}_t^{(d)} \geq 1 - n_t\right)$ – and the spot prices in the duration between such outliers are our data of interest.

We thus divide the data into a series of m timespans $\{\{\tau_0^{(j)}, \ldots, \tau_0^{(j)} + T_j\}\}_{j=1}^m$ where $\tau_0^{(j)}$ is the last time slot of a period when $\pi_t \geq \overline{\pi}$ occurs and $\tau_0^{(j)} + T_j$ is the next time slot of observing such an extreme price, which serves as the end of this timespan but the beginning of the next timespan, i.e., $\tau_0^{(j+1)} = \tau_0^{(j)} + T_i$ (the very first and last time slots are excluded). We use T_j to denote the length of the jth timespan, so we index the numbers of running on-demand instances n_t and active spot bids b_t within each timespan by $t = 0, 1, \ldots, T_j$. With the above notation, we can denote the observed spot prices by

$$\Pi = \left\{ \left\{ \pi_t^{(j)} \right\}_{t=0}^{T_j} \right\}_{j=1}^{m} \tag{8}$$

and the hidden states by

$$\mathcal{X} = \left\{ \left\{ \begin{pmatrix} n_t^{(j)} \\ b_t^{(j)} \end{pmatrix} \right\}_{t=0}^{T_j} \right\}_{j=1}^{m}, \tag{9}$$

where the states and prices during each timespan j are indicated by the superscript (j). Note that we must have $\forall\ j > 1$ that $\pi_0^{(j)} = \overline{\pi}$, $n_0^{(j)} = 1$ and $\forall\ j < m$ that $\pi_{T_j}^{(j)} = \overline{\pi}$, $n_{T_j}^{(j)} = 1$. Together with nonnegativity constraints on $n_t^{(j)}$ and $b_t^{(j)}$, we thus have the linear inequalities defining the feasible set of hidden states:

$$\begin{aligned} n_t^{(j)} &\geq 0\ \forall\ j,t & b_t^{(j)} &\geq 0\ \forall\ j,t \\ n_0^{(j)} &= 1\ \forall\ j > 1 & n_{T_j}^{(j)} &= 1\ \forall\ j < m. \end{aligned} \tag{10}$$

We solve the problem of jointly estimating the hidden states $\mathcal{X}$ and the model parameters Θ using the EM algorithm, which is a two step iterative algorithm that is guaranteed to converge to a (possibly sub-optimal) solution in terms of the expected log-likelihood [28]. In the E-step, we estimate the distribution over hidden states given the current estimates of the model parameters Θ; in the M-step, we use these estimated distributions to update the parameter estimates by maximizing the expected log-likelihood. If the estimated state distribution is given as an empirical distribution by sets of samples, the M-step parameter update admits a closed-form solution, as detailed in the Appendix of the technical report [11].

Estimating the distributions of these hidden states is a more challenging task. Although one can extend the Kalman filter to nonlinear dynamical systems via linearization, i.e., using the Extended Kalman Filter [29] or the Unscented Kalman Filter [30], they are likely to perform poorly on our model. The reason is that our spot price observation function (6) is non-smooth and possibly ill-posed (it tends to $-\infty$ as $b_t \to 0$), and the fact that we have constraints on the state space (10). Therefore, we need to resort to algorithms that support more modeling flexibility. We make use of Sequential Monte Carlo (SMC) methods (i.e. particle filters), which use sample trajectories (particle paths) over time to approximate the hidden state distributions; each sample is weighted by the probability that it generates the observed data [31].

While SMC has been most widely and successfully used in robotics, the low dimensionality of our model means that we only need a small number of particles to well-approximate the hidden state distributions. In Algorithm 1 we describe an SMC algorithm that takes as input an observed series of spot prices at each time slot t and returns K particles that are used to approximate the hidden states $\hat{X}_t^{(k)} \in \mathbb{R}^2$. The filter computes K trajectories in state space of length

Algorithm 1: Partial Trajectory Particle Filter

```
Data: spot prices $\pi_0, \dots, \pi_T$, model parameters $\Theta$,
      particle count $K$, trajectory length $L$, threshold $\epsilon$
Result: state distribution samples $\hat{X}_0^{(1)}, \dots, \hat{X}_T^{(K)} \in \mathbb{R}^2$
begin
    draw state $\hat{X}_0^{(k)} \in \mathbb{R}^2$ with probability $\mathbb{P}\left(\pi_0 \middle| \hat{X}_0^{(k)}\right)$
    set trajectory start time $t_0 = 0$
    while $t_0 < T$ do
        initialize probabilities $p^{(k)} \propto \mathbb{P}\left(\pi_{t_0} \middle| \hat{X}_{t_0}^{(k)}\right)$
        for $t = t_0 + 1, \dots, t_0 + 2L$ do
            sample state update $\hat{X}_t^{(k)} \middle| \hat{X}_{t-1}^{(k)}$ using (7)
            update probabilities $p^{(k)} \propto p^{(k)} \cdot \mathbb{P}\left(\pi_t \middle| \hat{X}_t^{(k)}\right)$
            if $p^{(k)} < \epsilon$ then
                with probability $p^{(k')}$ assign
                  $\hat{X}_{t_0:t}^{(k)} = \hat{X}_{t_0:t}^{(k')}$ and $p^{(k)} = p^{(k')}$
        with probability $p^{(k')}$ assign $\hat{X}_{t_0:t_0+L}^{(k)} = \hat{X}_{t_0:t_0+L}^{(k')}$
        set new trajectory start time $t_0 = t_0 + L$
```

L using the dynamical equations (7) and replaces those with negligible probability (below a threshold $\epsilon \ll 1$) of generating the observed data with more likely trajectories. To mitigate the sample degeneracy problem in SMC methods, in which eventually only one trajectory is left after resampling, we do not resample the full state history but only up to the past L time slots. Although in the general case this may cause low-probability transitions between trajectories, this is mitigated in our case by the (usually) slow evolution of the spot price.

Figure 2 shows the results of our spot price prediction algorithm and the associated hidden states n_t and b_t for two instance types in the g2 family. For particle filtering we use $K = 10^4$ particles, a trajectory length $L = 16$, and machine precision for ϵ. The left figures show the results for a less-volatile general purpose computing instance (m3.medium) while the right figures are those for a GPU instance (g2.2xlarge) with a highly variable spot price From the bottom two figures, we see that our predictions for both configurations closely track the actual spot prices; from the top figures, we observe that the hidden states also track changes in the spot prices. For instance, the spike around March 1 for m3.medium is accompanied by similarly dramatic changes in n_t and b_t. We further observe that g2.2xlarge has much greater fluctuation in the spot prices with corresponding higher fluctuations in the hidden states, possibly due to a smaller total capacity compared to exogenous demand. Overall these numerical results combined with the dicussion on full capacity-utilization in Section III-A suggest that much of the cloud is usually used by spot users.

IV. Bidding Strategies for Instance Families

For many spot users, simply bidding the on-demand price is a reasonable strategy; the spot price is usually lower than the on-demand price, so this limits the number of job interruptions while ensuring that completing the job costs less than in the on-demand market. However, this strategy does not necessarily minimize users' costs, particularly if users myopically bid on a given instance type instead of optimizing over other instances in the same family. A better strategy would be to consider the collective behavior of the spot prices over time [10], which we do in this section by accounting for their temporal dynamics.

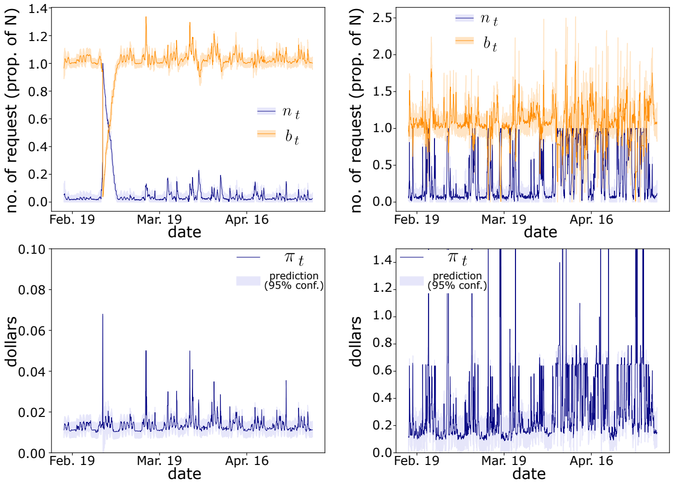


Fig. 2. Evolution of the hidden state estimates (top) and predicted spot price (bottom) for instance types m3.medium (left) and g2.2xlarge (right) in the US-East-1 region (zone a) over 3 months in 2017. Parameter estimates for m3.medium: $\theta_d = 98.6, \tilde{\theta}_d = 94.8, \theta_s = 79.1, \tilde{\theta}_s = 82.2, \sigma^2 = 3.15 \times 10^{-6}$; parameter estimates for instance type g2.2xlarge: $\theta_d = 13.0, \tilde{\theta}_d = 12.2, \theta_s = 11.4, \tilde{\theta}_s = 13.1, \sigma^2 = 2.83 \times 10^{-3}$. The larger values of the exponential distribution parameters (θ) for the first instance reflect the lower volatility of the m3.medium instance, which as can be seen from the bottom plots reaches its on-demand price ($0.067) much less than g2.2xlarge reaches its own ($0.65). The spot price observation noise (σ^2) is also much lower. The similarity of the arrival and departure parameters (θ v.s. $\tilde{\theta}$), which implies that requests arrive and jobs complete at roughly similar rates, will be important for the bidding strategies discussed in Section IV.

In this section, we propose bidding strategies that leverage the cloud dynamics model from Section III in a practical setting. We then compare the cost achieved by our approach to that of an auto-regressive baseline approach [9] and show superior performance across different instance families. Since minimizing the cost of a job is highly dependent on the ability to predict several time slots into the future, *these results are indicative of the superior predictive power of our model, compared to the auto-regressive baseline.*

A. Choosing Between Instances

We consider the following task. Suppose we have instance types indexed by $i \in [I]$ for $I > 1$, each with on-demand price $\overline{\pi}^{(i)}$, and a user at time τ with a job that runs in time T_i on instance i.[1] Throughout this section, we use the superscript (i) to index the instance. For simplicity, we assume that the user wishes to start a job immediately, incurs a negligible probability of job interruption (as can be assured by bidding

[1] Clearly, jobs on different instance types have different completion times. The exact relationship of time T_i on different instances depends on a job's particular resource requirements and the instances' resource configurations.

the on-demand price), and has a loose deadline, i.e., the job should be completed by some time $> \tau + \max_{i \in I} T_i$. The user then wishes to minimize the cost

$$P_\tau^{(i)} = \sum_{t=\tau+1}^{t+T_i} \pi_t^{(i)} \tag{11}$$

needed to complete his job. Given the spot price $\pi_\tau^{(i)}$ for each instance type i at time τ, we assume a conditional distribution on the future prices $\pi_{\tau+1}^{(i)}, \ldots, \pi_{\tau+T_i}^{(i)}$. The optimal strategy is then to choose instance

$$i_\tau^* = \arg\min_{i \in I} \quad \mathbb{E}_{\pi_\tau^{(i)}} \left(P_\tau^{(i)} \right) \tag{12}$$

and bid the on-demand price $\overline{\pi}^{(i^*)}$. Here the subscript $\pi_\tau^{(i)}$ denotes the conditional expectation.

This bidding scenario can readily be applied to a realistic setting in which Amazon EC2 offers a family of machines in the same region[2]. We define a "family" of instances as a set of instance types that emphasize a certain aspect of performance, such as compute optimized `c3.*` instances, memory optimized `r3.*` instances, and storage optimized `i3.*` instances. Instance types in each family offer different amounts of compute, memory, and storage resources in the same ratio, e.g., `micro`, `small`, `medium`, `large`, `xlarge`, and `2xlarge`. For example, the `m3.*` family of general computing instances includes types `medium`, `large`, `xlarge`, and `2xlarge`, offering 1, 2, 4, and 8 vCPUs, respectively. We can then suppose that jobs' running times increase linearly in the scale of the resources offered, e.g., that we have a perfectly parallelizable job that does not require more memory than available in any `m3.*` instance. If the job runs in time T on the `m3.medium` instance, it can be expected to run in times $\frac{T}{2}$, $\frac{T}{4}$, and $\frac{T}{8}$, respectively, on the other instance types in the `m3` family. The linear scaling of the on-demand price for these instances with the number of vCPUs (0.067, 0.133, 0.266, and 0.532 US cents, respectively) is consistent with this assumption [32].

More formally, consider a family of instance types indexed by $i \in [I]$ in ascending order by number of CPUs the instance contains, where $C_i \,\forall\, i$ denotes the scaling of instance i compared to instance 1. Then, if a perfectly parallelizable job runs in time T on instance 1, the cost minimization problem is to find an instance in the family yielding the least cost:

$$i_\tau^* = \arg\min_{i \in [I]} \quad \mathbb{E}_{\pi_\tau^{(i)}} \left(P_\tau^{(i)} \right) = \arg\min_{i \in [I]} \quad \sum_{t=\tau+1}^{\tau + \frac{T}{C_i}} \mathbb{E}_{\pi_\tau^{(i)}} \left(\pi_t^{(i)} \right) \tag{13}$$

We now consider three approaches to solving this problem given the spot prices $\pi_\tau^{(i)}$. The first two are based on the cloud dynamics model introduced in Section III, while the second is a baseline auto-regression model used for comparison.

Monte Carlo Simulation: After we have used EM as in Section III to learn parameter estimates $\hat{\Theta}_i$ for each instance i, we can directly estimate the distribution of the hidden states $n_\tau^{(i)}, b_\tau^{(i)}$ at time τ as in Algorithm 1 and then approximate the expectation of each future price $\mathbb{E}_{\pi_\tau^{(i)}} \left(\pi_t^{(i)} \right)$ by simple Monte Carlo simulation. (13) can then be solved by substituting these simulated expectations.

Taylor Approximation: While the above approach most closely follows the cloud dynamics model, under reasonable assumptions on the parameters Θ_i that can be empirically observed, we can construct a second-order approximation to $\mathbb{E}_{\pi_\tau^{(i)}} \left(\pi_t^{(i)} \right)$ that is less noisy and less computationally complex than the Monte Carlo approach:

Lemma 2: Suppose for some instance i (where with slight abuse of notation, we drop the superscript (i) for simplification) the parameters of the arrival-departure model satisfy $\theta_d \approx \tilde{\theta}_d$ and $\theta_s \approx \theta \approx \tilde{\theta}_s$ and that the spot price π_τ is in the interval $(\underline{\pi}, \overline{\pi})$. If we have unbiased estimates $\bar{n}_\tau$ of n_τ, $\bar{b}_\tau$ of b_τ, and S_τ^2 of $(b_\tau - \mathbb{E}_{\pi_\tau}(b_\tau))^2$ then for small enough δ we can approximate the expected spot price at time $t = \tau + \delta$ by

$$\mathbb{E}_{\pi_\tau}(\pi_t) \approx \overline{\pi} - (\overline{\pi} - \underline{\pi}) \frac{1 - \bar{n}_\tau}{\bar{b}_\tau} \left(1 + \frac{S_\tau^2 + \frac{2\delta}{\theta^2}}{\bar{b}_\tau^2} \right) \tag{14}$$

The unbiased estimates of $\bar{n}_\tau, \bar{b}_\tau$, and S_τ^2 can be found by drawing samples using the approach in Algorithm 1 and taking the first and second sample moments. Furthermore, as noted in Figure 2, the values of θ and $\tilde{\theta}$ that parameterize the arrival and departure rates, respectively, in the model were found to be roughly the same for tested instances, allowing the approximation for this bidding strategy to be made.

Since we do not want to bid below the lower bound $\underline{\pi}$, when estimating the expected payment we correct (14) by taking the maximum of it and $\underline{\pi}$, yielding the following approximation:

$$\mathbb{E}_{\pi_\tau^{(i)}} \left(P_\tau^{(i)} \right) \approx \sum_{t=\tau+1}^{t + \frac{T}{C_i}} \max \left\{ \underline{\pi}^{(i)}, \mathbb{E}_{\pi_\tau^{(i)}} \left(\pi_t^{(i)} \right) \right\} \tag{15}$$

This approximation can then be substituted into (13) in order to easily find the lowest-cost instance type to pick.

Linear Auto-Regression: Linear auto-regression (AR) models assume that at all times t, the spot price $\pi_t^{(i)}$ is a linear function of the previous price points along with a mean-zero independent noise term. In our case, given $\pi_\tau^{(i)}$ the next step forecast is

$$\pi_{\tau+1}^{(i)} = c^{(i)} + \phi^{(i)} \pi_\tau^{(i)} + \varepsilon_t^{(i)} \tag{16}$$

where $c^{(i)}$ and $\phi^{(i)}$ are learnable parameters and $\varepsilon_t^{(i)}$ is white noise. Using this approach, the expected spot price $\mathbb{E}_{\pi_\tau^{(i)}} \left(\pi_{\tau+\delta}^{(i)} \right)$ at time $\tau + \delta$ can be found by iteratively applying (16) (removing the noise term) to successive estimates $\pi_\tau, \ldots, \pi_{\tau+\delta-1}$, and the expected payment $\mathbb{E}_{\pi_\tau^{(i)}} \left(P_{\frac{T}{C_i}}^{(i)} \right)$ is then the sum of the resulting estimates.

Although the assumptions of this AR model do not hold when the model is applied to spot price data (see Figure 3), the

[2] Bidding across regions can lead to high communication latency, which is out of the scope of this paper. If we neglect communication latency, we can optimize across regions by finding the optimal bid and associated expected payment in each region and choosing the one with the least expected payment.

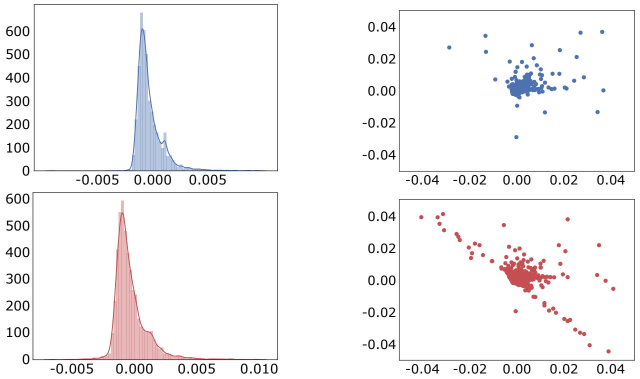


Fig. 3. Plots of residual distribution (left) and auto-correlation (right) of an AR(1) (top) and an AR(17) (bottom), where the latter is chosen via the Akaike Information Criterion [33]. Both models were fit on `m3.medium` data from the US-East-1 region (zone a). The distributions are clearly skewed and there is correlation between residuals at different timesteps, thus violating two main assumptions on the errors in auto-regression models.

model is widely used, including to predict spot prices (notably in [9]) and thus serves as useful baseline for comparison.

B. Bidding Strategy Evaluation

We evaluate each of the three bidding strategies introduced in the previous section on 160 days of spot price data. First we train a separate model (either the one in Section III or an AR model) for each instance i in a family of instance $[I]$ on the first half of the spot price data. Then at each timestep τ in the second half of the data, we solve (13) by computing the expected payment needed to finish the job for each instance in the family and taking the one with the lowest value (i.e. i_τ^*). Then to compare the cost of the selected instance with that of the best strategy to use in hindsight we look $\frac{T}{C_i}$ steps ahead and compute the *a posteriori* payment (11) for each instance type i. Finally, we compare the job completion cost of our bidding strategy compared to that of the best strategy by computing the regret

$$R_\tau = P_\tau^{(i_\tau^*)} - \min_{i\in[I]} P_\tau^{(i)} = \sum_{t=\tau+1}^{T_{i_\tau^*}} \pi_t^{(i_\tau^*)} - \min_{i\in[I]} \sum_{t=\tau+1}^{T_i} \pi_t^{(i)}. \quad (17)$$

Note that $R_\tau = 0$ when our bidding strategy succesfully selects the optimal instance type to use hindsight, i.e. when $i_\tau^* = \arg\min_{i\in[I]} P_\tau^{(i)}$, and is positive otherwise.

We evaluate all three bidding strategies on five popular instance familes and report the average regret over the test data in Table II, along with plots of relative error for several settings in Figure 4. From the results we observe that either Monte Carlo simulation or Taylor approximation almost always outperforms the auto regressive baseline (i.e., yields lower average regret), especially on longer timescales (bigger T). The poor forecasting performance on many instance families by the auto-regression model confirms the analysis in Figure 3 that it is a poor approximation for the evolution of spot prices.

Our evaluation considers jobs with lowest-configuration runtimes $T = 8$, 16, 32, and 64. As expected, regret increases with larger T, when overall payments are also higher. More notably, we find that while Monte Carlo simulation performs well at lower T, at higher T it is usually better to use the Taylor approximation strategy. This is likely because probabilistic simulation becomes more noisy with each additional timestep. Since longer timescales provide more return due to strategic bidding and the Taylor approximation approach requires less computation, it is likely to be preferred for most users.

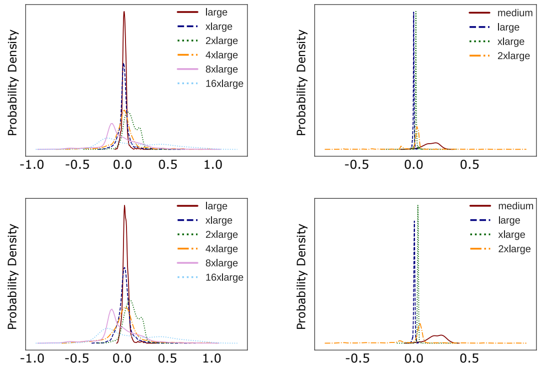


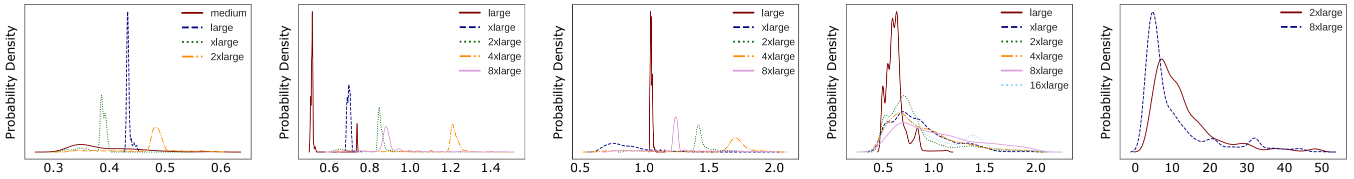
Fig. 4. Relative error of payment estimate on 80 days of spot price data for `i3.*` (left) and `m3.*` (right) families and $T = 8$ (top) and $T = 16$ (bottom). Note that despite the spot price of the former family being much more volatile the error is still quite small and sharply concentrated around zero.

TABLE II
AVERAGE REGRET (IN US CENTS). THE LOWEST REGRET IN EACH INSTANCE FAMILY FOR EACH JOB LENGTH IS BOLDED.

T*	Bidding Strategy	Instance Family				
		`m3`	`c3`	`r3`	`i3`	`g2`
8	Monte Carlo Simulation	**0.68**	0.25	**0.18**	0.21	**11.09**
	Taylor Approximation	0.97	**0.20**	0.59	0.26	12.91
	Auto-Regression [9]	0.95	0.25	0.33	**0.18**	17.2
16	Monte Carlo Simulation	**1.49**	0.43	**0.40**	**0.50**	**26.20**
	Taylor Approximation	1.89	**0.39**	1.11	0.55	52.40
	Auto-Regression [9]	1.89	0.43	1.00	0.51	77.64
32	Monte Carlo Simulation	**2.95**	0.55	**1.28**	1.53	78.94
	Taylor Approximation	3.76	**0.54**	2.42	**0.87**	**69.88**
	Auto-Regression [9]	3.76	0.55	1.89	1.52	328.10
64	Monte Carlo Simulation	**5.86**	**0.49**	9.78	4.50	234.02
	Taylor Approximation	7.32	0.53	4.77	**2.23**	**180.85**
	Auto-Regression [9]	7.32	**0.49**	**3.11**	2.72	663.39

* Time (in hours) required to run the job to completion on the lowest-configuration instance in the family.

To gain a deeper understanding of why our model performs similarly to auto-regression on some instance families (e.g. `c3.*`) but consistently better on others (e.g. `i3.*`,`g2.*`), we plot in Figure 5 kernel density estimates of the distributions over all time of the realized payment for each instance in each of the five families. We see that instance families on which our bidding strategies perform especially well are those with significant overlap in the payment distributions of different instances, in which case there is no single instance type that consistently yields the lowest cost in that family. Unsurprisingly, these instance families also tend to be those whose spot price dynamics are more volatile (see for example



(a) `m3.*`. (b) `c3.*`. (c) `r3.*`. (d) `i3.*`. (e) `g2.*`.

Fig. 5. Distribution of payments for different instance types in five instance families.

the history of `g2.2xlarge` in Figure 2), which are thus also more likely to experience price fluctuations that allow for strategic selection between instances. The fact that our model is especially good at payment prediction in such cases, achieiving 1.7 and 4.7 times lower regret on `i.3*` and `g.2*`, respectively, is further indication of the forecasting power of the cloud dynamics model as well as the importance of accurate spot price predictions for optimizing user cost, particularly when the spot prices are volatile.

V. Conclusion

Cloud spot pricing, as a mechanism for CPs to auction off spare resources at highly reduced prices, creates opportunities not only for CPs to improve resource utilization but also for users to minimize the cost of their jobs with sophisticated bidding strategies. In this work, we first developed a nonlinear dynamical system model to capture the dynamics of spot pricing, by formulating the spot price as a function of latent states characterizing user demands in both on-demand and spot markets. Our model enables us to predict future spot prices conditioned on observations of past spot prices.

Then, we designed user bidding strategies by leveraging these predictions and trading off between completion time and price dynamics across all instances in the same instance family. These strategies can further lower user costs, as compared to bidding on a single instance type. We validated our dynamic spot price model on 80 days of Amazon EC2's spot price data and simulated our bidding strategies on another 80 days of data. An analysis of our bidding strategies shows that for difficult-to-predict volatile instances we are able to a 4 times larger user cost reduction than auto-regressive forecasting.

References

[1] Google Cloud Platform, "Compute Engine Pricing," 2017. https://cloud.google.com/compute/.

[2] Amazon EC2, "Reserved Instances," 2017. https://aws.amazon.com/ec2/purchasing-options/reserved-instances/.

[3] Amazon EC2, "Spot Instances," 2017. https://aws.amazon.com/ec2/spot/.

[4] F. R. Dogar, T. Karagiannis, H. Ballani, and A. Rowstron, "Decentralized task-aware scheduling for data center networks," in *Proc. of ACM SIGCOMM*, 2014.

[5] Z. Huang, S. M. Weinberg, L. Zheng, C. Joe-Wong, and M. Chiang, "Discovering valuations and enforcing truthfulness in a deadline-aware scheduler," in *Proc. of IEEE INFOCOM*, 2017.

[6] X. Meng, V. Pappas, and L. Zhang, "Improving the scalability of data center networks with traffic-aware virtual machine placement," in *Proc. of IEEE INFOCOM*, 2010.

[7] J. Xu and J. A. Fortes, "Multi-objective virtual machine placement in virtualized data center environments," in *Proc. of IEEE/ACM GREENCOM-CPSCOM*, 2010.

[8] Google Cloud Platform, "Preemptible Virtual Machines," 2017. https://cloud.google.com/preemptible-vms/.

[9] O. Agmon Ben-Yehuda, M. Ben-Yehuda, A. Schuster, and D. Tsafrir, "Deconstructing Amazon EC2 spot instance pricing," *ACM Trans. on Economics and Computation*, vol. 1, no. 3, p. 16, 2013.

[10] L. Zheng, C. Joe-Wong, C. W. Tan, M. Chiang, and X. Wang, "How to bid the cloud," in *Proc. of ACM SIGCOMM*, 2015.

[11] M. Khodak, L. Zheng, A. S. Lan, C. Joe-Wong, and M. Chiang, "Learning and leveraging spot price dynamics for bidding strategy optimization: Technical report," 2018. https://1drv.ms/b/s!Aise2pW0Yvrda7t-8QQHaJiJxDc.

[12] B. Javadi, R. K. Thulasiramy, and R. Buyya, "Statistical modeling of spot instance prices in public cloud environments," in *Proc. of IEEE UCC*, 2011.

[13] S. Wee, "Debunking real-time pricing in cloud computing," in *Proc. of IEEE/ACM CCGrid*, 2011.

[14] Z. Amekraz and M. Y. Hadi, "Prediction of Amazon spot price based on chaos theory using ANFIS model," in *Proc. of IEEE/ACS AICCSA*, 2016.

[15] P. Sharma, D. Irwin, and P. Shenoy, "How not to bid the cloud," in *Proc. of HotCloud*, 2016.

[16] M. Zafer, Y. Song, and K.-W. Lee, "Optimal bids for spot VMs in a cloud for deadline constrained jobs," in *Proc. of IEEE CLOUD*, 2012.

[17] M. Lumpe, M. B. Chhetri, Q. B. Vo, and R. Kowalczyk, "On estimating minimum bids for Amazon EC2 spot instances," in *Proc. of IEEE/ACM CCGrid*, 2017.

[18] A. Andrzejak, D. Kondo, and S. Yi, "Decision model for cloud computing under SLA constraints," in *Proc. of IEEE MASCOTS*, 2010.

[19] M. Taifi and J. Y. Shi, "Performance and reliability effects of multi-tier bidding on MapReduce in auction-based clouds," in *Proc. of IEEE SOSE*, 2013.

[20] X. Ouyang, D. Irwin, and P. Shenoy, "Spotlight: An information service for the cloud," in *Proc. of IEEE ICDCS*, 2016.

[21] I. Menache, O. Shamir, and N. Jain, "On-demand, spot, or both: Dynamic resource allocation for executing batch jobs in the cloud.," in *Proc. of IEEE ICAC*, 2014.

[22] X. Yi, F. Liu, Z. Li, and H. Jin, "Flexible instance: Meeting deadlines of delay tolerant jobs in the cloud with dynamic pricing," in *Proc. of IEEE ICDCS*, 2016.

[23] X. Zhang, Z. Huang, C. Wu, Z. Li, and F. Lau, "Online auctions in IaaS clouds: Welfare and profit maximization with server costs," in *Proc. of ACM SIGMETRICS*, 2015.

[24] X. Zhang, C. Wu, Z. Li, and F. C. Lau, "A truthful (1-ε)-optimal mechanism for on-demand cloud resource provisioning," in *Proc. of IEEE INFOCOM*, 2015.

[25] G. Sakellari and G. Loukas, "A survey of mathematical models, simulation approaches and testbeds used for research in cloud computing," *Simulation Modelling Practice and Theory*, vol. 39, pp. 92–103, 2013.

[26] L. Zheng, C. Joe-Wong, C. G. Brinton, C. W. Tan, S. Ha, and M. Chiang, "On the viability of a cloud virtual service provider," in *Proc. of ACM SIGMETRICS*, 2016.

[27] Y. Yao, L. Huang, A. Sharma, L. Golubchik, and M. Neely, "Data centers power reduction: A two time scale approach for delay tolerant workloads," in *Proc. of IEEE INFOCOM*, 2012.

[28] A. P. Demster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[29] Z. Ghahramani and S. T. Roweis, "Learning nonlinear dynamical systems using an EM algorithm," in *Proc. of NIPS*, 1999.

[30] E. A. Wan and R. van der Merwe, "The unscented Kalman filter for nonlinear estimation," in *Proc. of IEEE AS-SPCC*, 2000.

[31] S. Thrun, "Particle filters in robotics," in *Proc. of UAI*, 2002.

[32] A. EC2, "Pricing for amazon emr and amazon ec2 (on-demand)," 2017. https://aws.amazon.com/emr/pricing.

[33] H. Akaike, "Information theory and an extension of the maximum likelihood principle," in *Proc. of ISIT*, 1973.

Appendix

Throughout the appendix we will be assuming the cloud provider model as described in Section III. We define the indicator function $1_{b\geq\pi} = 1 \iff b \geq \pi$ and will make use of the following identity that follows from (2):

$$N_t^{(s)} = |\{b \in \mathcal{B}_t : b \geq \pi_t\}| = \sum_{b\in\mathcal{B}_t} 1_{b\geq\pi}$$

A. Proof of Proposition 1

Proof: By Definition 1 we have $\forall\ b \in \mathcal{B}_t$ and any π that if $b' \sim \mathcal{U}[\underline{\pi}, \overline{\pi}]$ then

$$\mathbb{P}\left(1_{b\geq\pi} = 1\right) = 1 - \mathbb{P}\left(b < \pi\right) \geq 1 - \mathbb{P}\left(b' < \pi\right) = \mathbb{P}\left(1_{b'\geq\pi} = 1\right)$$

and so it follows that

$$\mathbb{P}\left((\pi_t - \underline{\pi}) \sum_{b\sim\mathcal{U}^{B_t}[\underline{\pi},\overline{\pi}]} 1_{b\geq\pi_t} \geq x\right) \leq \mathbb{P}\left((\pi_t - \underline{\pi}) \sum_{b\in\mathcal{B}_t} 1_{b\geq\pi_t} \geq x\right)$$

for any π_t and x. Combining this with the fact that $B_t = |\mathcal{B}_t| \geq \sum_{b\in\mathcal{B}_t} 1_{b\geq\pi^{(\rho)}}$ yields

$$\begin{aligned}
\mathbb{P}\left(\pi_t \leq \pi^{(\rho)}\right) &= \mathbb{P}\left(\operatorname*{arg\,max}_{N_t^{(d)}+N_t^{(s)}\leq N} (\pi_t - \underline{\pi})N_t^{(s)} \leq \pi^{(\rho)}\right) \\
&= \mathbb{P}\left(\max_{\substack{N_t^{(d)}+N_t^{(s)}\leq N \\ \pi_t\leq\pi^{(\rho)}}} (\pi_t - \underline{\pi}) \sum_{b\in\mathcal{B}_t} 1_{b\geq\pi_t} \geq \max_{\substack{N_t^{(d)}+N_t^{(s)}\leq N \\ \pi_t>\pi^{(\rho)}}} (\pi_t - \underline{\pi}) \sum_{b\in\mathcal{B}_t} 1_{b\geq\pi_t}\right) \\
&\leq \mathbb{P}\left(\max_{\substack{N_t^{(d)}+N_t^{(s)}\leq N \\ \pi_t\leq\pi^{(\rho)}}} (\pi_t - \underline{\pi}) \sum_{b\in\mathcal{B}_t} 1_{b\geq\pi_t} \geq \max_{\substack{N_t^{(d)}+N_t^{(s)}\leq N \\ \pi_t>\pi^{(\rho)}}} (\pi_t - \underline{\pi}) \sum_{b\sim\mathcal{U}^{B_t}[\underline{\pi},\overline{\pi}]} 1_{b\geq\pi_t}\right) \\
&\leq \mathbb{P}\left((\pi^{(\rho)} - \underline{\pi}) \min\left\{B_t, N - N_t^{(d)}\right\} \geq \left(\pi^{\left(\frac{1}{2}\right)} - \underline{\pi}\right) \min\left\{\sum_{b\sim\mathcal{U}^{B_t}[\underline{\pi},\overline{\pi}]} 1_{b\geq\pi^{\left(\frac{1}{2}\right)}}, N - N_t^{(d)}\right\}\right) \\
&= \mathbb{P}\left((\pi^{(\rho)} - \underline{\pi})B_t \geq (\pi^{\left(\frac{1}{2}\right)} - \underline{\pi}) \sum_{b\sim\mathcal{U}^{B_t}[\underline{\pi},\overline{\pi}]} 1_{b\geq\pi^{\left(\frac{1}{2}\right)}} \middle| X_t\right) \mathbb{P}\left(X_t\right) \\
&\quad + \mathbb{P}\left((\pi^{(\rho)} - \underline{\pi})\left(N - N_t^{(d)}\right) \geq (\pi^{\left(\frac{1}{2}\right)} - \underline{\pi}) \sum_{b\sim\mathcal{U}^{B_t}[\underline{\pi},\overline{\pi}]} 1_{b\geq\pi^{\left(\frac{1}{2}\right)}} \middle| Y_t\right) \mathbb{P}\left(Y_t\right) \\
&\quad + \mathbb{P}\left((\pi^{(\rho)} - \underline{\pi})\left(N - N_t^{(d)}\right) \geq (\pi^{\left(\frac{1}{2}\right)} - \underline{\pi})(N - N_t^{(d)}) \middle| Z_t\right) \mathbb{P}\left(Z_t\right)
\end{aligned}$$

for X_t the event that $N - N_t^{(d)} \geq B_t$, Y_t the event that $\sum_{b\sim\mathcal{U}^{B_t}[\underline{\pi},\overline{\pi}]} 1_{b\geq\pi^{\left(\frac{1}{2}\right)}} \leq N - N_t^{(d)} < B_t$, and Z_t the event that $N - N_t^{(d)} < \sum_{b\sim\mathcal{U}^{B_t}[\underline{\pi},\overline{\pi}]} 1_{b\geq\pi^{\left(\frac{1}{2}\right)}}$. Now the last conditional probability is zero since $\pi^{(\rho)} - \underline{\pi} < \pi^{\left(\frac{1}{2}\right)} - \underline{\pi}$. Then from Y_t we have $B_t > N - N_t^{(d)}$ so the second conditional probability is bounded by the first. Combining terms and applying Hoeffding's inequality gives

$$\begin{aligned}
\mathbb{P}\left(\pi_t \leq \pi^{(\rho)}\right) &\leq \mathbb{P}\left((\pi^{(\rho)} - \underline{\pi})B_t \geq (\pi^{\left(\frac{1}{2}\right)} - \underline{\pi}) \sum_{b\sim\mathcal{U}^{B_t}[\underline{\pi},\overline{\pi}]} 1_{b\geq\pi^{\left(\frac{1}{2}\right)}}\right) \\
&= \mathbb{P}\left(2\rho B_t \geq \sum_{b\sim\mathcal{U}^{B_t}[\underline{\pi},\overline{\pi}]} 1_{b\geq\pi^{\left(\frac{1}{2}\right)}}\right) \\
&\leq \exp\left(-2\left(\frac{1}{2} - 2\rho\right)^2 B_t\right)
\end{aligned}$$

■

B. Proof of Lemma 1

Proof: Given $N_t^{(d)}$ and B_t, define the random variables $X_t = \frac{1}{B_t} \sum\limits_{b \in \mathcal{B}_t} 1_{b \geq \pi}$ and $Y_t = \frac{1}{B_t} \sum\limits_{b \in \mathcal{B}_t} \frac{\overline{\pi} - \pi}{\overline{\pi} - \underline{\pi}} - \frac{1}{2} + 1_{b \geq \frac{\overline{\pi} + \underline{\pi}}{2}}$ for $\pi \in [\underline{\pi}, \overline{\pi}]$, and let f_{X_t} and f_{Y_t} be their respective probability density functions. Now since

$$\mathbb{E}X_t = \mathbb{E}1_{b \geq \pi} = \frac{\overline{\pi} - \pi}{\overline{\pi} - \underline{\pi}} = \mathbb{E}\left(\frac{\overline{\pi} - \pi}{\overline{\pi} - \underline{\pi}} - \frac{1}{2} + 1_{b \geq \frac{\overline{\pi} + \underline{\pi}}{2}} \right) = \mathbb{E}Y_t$$

we have by the Strong Law of Large Numbers that X_t and Y_t converge almost surely to the same limit as $B_t \to \infty$. Therefore using the Bounded Convergence Theorem we have that $\forall\ x$

$$\lim_{B_t \to \infty} \mathbb{P}(X_t \leq x) = \int\limits_0^x \lim_{B_t \to \infty} f_{X_t} = \int\limits_0^x \lim_{B_t \to \infty} f_{Y_t} = \lim_{B_t \to \infty} \mathbb{P}(Y_t \leq x)$$

Following Equation 5 and using the above identity we have

$$\begin{aligned}
\lim_{\substack{B_t \to \infty \\ \frac{N - N_t^{(d)}}{B_t} = C_t}} \mathbb{P}\left(\pi_t \leq \pi \middle| N_t^{(d)}, B_t \right) &= \lim_{\substack{B_t \to \infty \\ \frac{N - N_t^{(d)}}{B_t} = C}} \mathbb{P}\left(\sum_{b \in \mathcal{B}_t} 1_{b \geq \pi} \leq N - N_t^{(d)} \middle| N_t^{(d)}, B_t \right) \\
&= \lim_{B_t \to \infty} \mathbb{P}\left(X_t \leq C_t \middle| N_t^{(d)}, B_t \right) \\
&= \lim_{B_t \to \infty} \mathbb{P}\left(Y_t \leq C_t \middle| N_t^{(d)}, B_t \right) \\
&= \lim_{B_t \to \infty} \mathbb{P}\left(\frac{\overline{\pi} - \pi}{B_t} \sum_{b \in \mathcal{B}_t} Z_t^{(b)} \leq \pi \middle| N_t^{(d)}, B_t \right)
\end{aligned}$$

where the random variable $Z_t^{(b)} = 1_{b \geq \frac{\overline{\pi} + \underline{\pi}}{2}} + \frac{\overline{\pi} - \underline{\pi}}{2(\overline{\pi} - \underline{\pi})} - C_t$ has expectation $\mathbb{E}Z_t^{(b)} = \frac{\overline{\pi}}{\overline{\pi} - \underline{\pi}} - C_t$ and variance $\text{Var}\left(Z_t^{(b)} \right) = \frac{1}{4}$. Thus as $B_t \to \infty$ the Central Limit Theorem implies that $\frac{1}{B_t} \sum\limits_{b \in \mathcal{B}_t} Z_t^{(b)}$ converges in distribution to $\mathcal{N}\left(\frac{\overline{\pi}}{\overline{\pi} - \underline{\pi}} - C_t, \frac{1}{4B_t} \right)$. Then it follows from the previous equalities and the equivalence between convergence in distribution and pointwise convergence of the cumulative density function that

$$\lim_{\substack{B_t \to \infty \\ \frac{N - N_t^{(d)}}{B_t} = C_t}} \mathbb{P}\left(\pi_t \leq \pi \middle| N_t^{(d)}, B_t \right) = \lim_{\substack{B_t \to \infty \\ \frac{N - N_t^{(d)}}{B_t} = C_t}} \mathbb{P}\left(Z_t \leq \pi \middle| N_t^{(d)}, B_t \right)$$

$$Z_t \sim \mathcal{N}\left(\overline{\pi} - (\overline{\pi} - \underline{\pi}) \frac{N - N_t^{(d)}}{B_t}, \frac{(\overline{\pi} - \underline{\pi})^2}{4B_t} \right)$$

Rewriting this last expression in terms of the normalizations n_t and b_t of $N_t^{(d)}$ and B_t completes the proof. ∎

C. Estimating Parameters via Expectation-Maximization

Given the spot price data Π (8) and the inferred hidden states $\mathcal{X}$ (9), the model likelihood given the parameters Θ is expressed as

$$\mathcal{L}_\Theta (\Pi, \mathcal{X}) = \prod_{j=1}^{m} \prod_{t=1}^{T_j} f_{\theta_d, \tilde{\theta}_d} \left(n_t^{(j)} - n_{t-1}^{(j)} \right) f_{\theta_s, \tilde{\theta}_s} \left(b_t^{(j)} - b_{t-1}^{(j)} \right) \prod_{\substack{t = 1_{j > 1} \\ n_t^{(j)} + b_t^{(j)} > 1}}^{T_j - 1_{j < m}} g_{\frac{\sigma^2}{b_t^{(j)}}} \left(\pi_t^{(j)} - \overline{\pi} + (\overline{\pi} - \underline{\pi}) \frac{1 - n_t^{(j)}}{b_t^{(j)}} \right)$$

where $f_{\mu,\nu}(x)$ is the probability density function of the difference of random variables drawn from exponential distributions with parameters μ, ν, which is expressed as

$$f_{\mu,\nu}(x) = \frac{\mu\nu}{\mu + \nu} \begin{cases} \exp(\nu x) & x < 0 \\ \exp(-\mu x) & x \geq 0 \end{cases}$$

and g_Σ is the probability density function of a normally distributed random variable with mean zero and covariance Σ. Note that the index restrictions placed above and below the last product sign in the likelihood are to account for states for which the spot price conditioned on the states satisfies $\pi_t^{(j)} \in \{\underline{\pi}, \overline{\pi}\}$ with probability 1 (following (6)) and which can thus be excluded from the product.

Now since in EM we only know the (approximate) distribution over the hidden states, we will be maximizing the expected log-likelihood $\mathbb{E}_{\mathcal{X}} \log \mathcal{L}_{\Theta}\left(\Pi, \mathcal{X}\right)$. We approximate this expectation by taking the average of the log of the above expression over K samples $n_t^{(j,k)}$ and $b_t^{(j,k)}$ returned by SMC. We apply first-order optimality conditions to find the system solved by the parameter estimates:

$$
\begin{aligned}
0 &= \frac{\mathcal{T}}{\theta_d} - \frac{\mathcal{T}}{\theta_d + \tilde{\theta}_d} - \frac{1}{K} \sum_{k=1}^{K} \sum_{j=1}^{m} \sum_{t=1}^{T_j} \max\left\{0, n_t^{(j,k)} - n_{t-1}^{(j,k)}\right\} \\
0 &= \frac{\mathcal{T}}{\tilde{\theta}_d} - \frac{\mathcal{T}}{\theta_d + \tilde{\theta}_d} + \frac{1}{K} \sum_{k=1}^{K} \sum_{j=1}^{m} \sum_{t=1}^{T_j} \min\left\{0, n_t^{(j,k)} - n_{t-1}^{(j,k)}\right\} \\
0 &= \frac{\mathcal{T}}{\theta_s} - \frac{\mathcal{T}}{\theta_s + \tilde{\theta}_s} - \frac{1}{K} \sum_{k=1}^{K} \sum_{j=1}^{m} \sum_{t=1}^{T_j} \max\left\{0, b_t^{(j,k)} - b_{t-1}^{(j,k)}\right\} \\
0 &= \frac{\mathcal{T}}{\tilde{\theta}_s} - \frac{\mathcal{T}}{\theta_s + \tilde{\theta}_s} + \frac{1}{K} \sum_{k=1}^{K} \sum_{j=1}^{m} \sum_{t=1}^{T_j} \min\left\{0, b_t^{(j,k)} - b_{t-1}^{(j,k)}\right\} \\
0 &= \frac{(\mathcal{T} - m + 2)K - R}{K\sigma} - \frac{1}{K\sigma^3} \sum_{k=1}^{K} \sum_{j=1}^{m} \sum_{\substack{t=1_{j>1} \\ n_t^{(j,k)} + b_t^{(j,k)} > 1}}^{T_j - 1_{j<m}} \left(\pi_t^{(j)} - \overline{\pi} + (\overline{\pi} - \underline{\pi}) \frac{1 - n_t^{(j,k)}}{b_t^{(j,k)}}\right)^2 b_t^{(j,k)}
\end{aligned}
$$

where $\mathcal{T} = \sum_{j=1}^{m} T_j$ and $R = \left|\left\{(j,k) : n_t^{(j,k)} + b_t^{(j,k)} \leq 1\right\}\right|$. Solving this system for $\Theta = \{\theta_d, \tilde{\theta}_d, \theta_s, \tilde{\theta}_s, \sigma^2\}$ yields the following parameter estimates:

$$
\hat{\theta}_d = \frac{\mathcal{T}K}{\sqrt{n^+}\left(\sqrt{n^+} + \sqrt{n^-}\right)} \qquad \hat{\tilde{\theta}}_d = \frac{\mathcal{T}K}{\sqrt{n^-}\left(\sqrt{n^+} + \sqrt{n^-}\right)}
$$

$$
\hat{\theta}_s = \frac{\mathcal{T}K}{\sqrt{b^+}\left(\sqrt{b^+} + \sqrt{b^-}\right)} \qquad \hat{\tilde{\theta}}_s = \frac{\mathcal{T}K}{\sqrt{b^-}\left(\sqrt{b^+} + \sqrt{b^-}\right)}
$$

$$
\hat{\sigma}^2 = \frac{\sum_{k=1}^{K} \sum_{j=1}^{m} \sum_{\substack{t=1_{j>1} \\ n_t^{(j,k)} + b_t^{(j,k)} > 1}}^{T_j - 1_{j<m}} \left(\pi_t^{(j)} - \overline{\pi} + (\overline{\pi} - \underline{\pi}) \frac{1 - n_t^{(j,k)}}{b_t^{(j,k)}}\right)^2 b_t^{(j,k)}}{(\mathcal{T} - m + 2)K - R}
$$

where

$$
n^+ = \sum_{k=1}^{K} \sum_{j=1}^{m} \sum_{t=1}^{T_j} \max\left\{0, n_t^{(j,k)} - n_{t-1}^{(j,k)}\right\} \qquad n^- = -\sum_{k=1}^{K} \sum_{j=1}^{m} \sum_{t=1}^{T_j} \min\left\{0, n_t^{(j,k)} - n_{t-1}^{(j,k)}\right\}
$$

$$
b^+ = \sum_{k=1}^{K} \sum_{j=1}^{m} \sum_{t=1}^{T_j} \max\left\{0, b_t^{(j,k)} - b_{t-1}^{(j,k)}\right\} \qquad b^- = -\sum_{k=1}^{K} \sum_{j=1}^{m} \sum_{t=1}^{T_j} \min\left\{0, b_t^{(j,k)} - b_{t-1}^{(j,k)}\right\}
$$

The approximately optimal expected log-likelihood is thus

$$
\begin{aligned}
\mathbb{E}_{\mathcal{X}} \log \mathcal{L}_{\hat{\Theta}}\left(\Pi, \mathcal{X}\right) = \mathcal{T} \log\left(\frac{\hat{\theta}_d \hat{\tilde{\theta}}_d}{\hat{\theta}_d + \hat{\tilde{\theta}}_d} \frac{\hat{\theta}_s \hat{\tilde{\theta}}_s}{\hat{\theta}_s + \hat{\tilde{\theta}}_s}\right) &- \frac{\hat{\theta}_d n^+ + \hat{\tilde{\theta}}_d n^- + \hat{\theta}_s b^+ + \hat{\tilde{\theta}}_s b^-}{K} \\
&- \frac{(\mathcal{T} - m + 2)K - R}{2K}\left(1 + \log 2\pi\hat{\sigma}^2\right) + \frac{1}{2K} \sum_{k=1}^{K} \sum_{j=1}^{m} \sum_{\substack{t=1_{j>1} \\ n_t^{(j,k)} + b_t^{(j,k)} > 1}}^{T_j - 1_{j<m}} \log b_t^{(j,k)}
\end{aligned}
$$

D. *Proof of Lemma 2*

Proof: Under the assumptions $\theta_d \approx \tilde{\theta}_d$ and $\theta_s \approx \theta \approx \tilde{\theta}_s$ we have that $\eta^{(d)}_{t'} = \lambda^{(d)}_{t'} - \tilde{\lambda}^{(d)}_{t'}$ and $\eta^{(s)}_{t'} = \lambda^{(s)}_{t'} - \tilde{\lambda}^{(s)}_{t'}$ approximately follow zero-mean Laplace distributions at all times t', with the distribution of $\eta^{(s)}_{t'}$ having scale parameter $\frac{1}{\theta}$. Thus by applying Equation 7 repeatedly and using the fact that $\bar{n}_\tau$ and $\bar{b}_\tau$ are unbiased estimators of n_τ and b_τ, respectively, we have that

$$\mathbb{E}\left(\bar{n}_\tau - n_t\right) = \mathbb{E}\left(\bar{n}_\tau - n_\tau - \sum_{t'=\tau+1}^{t} \lambda^{(d)}_{t'} - \tilde{\lambda}^{(d)}_{t'}\right) = \mathbb{E}\left(\bar{n}_\tau - n_\tau\right) + \sum_{t'=\tau+1}^{t} \mathbb{E}\left(\eta^{(d)}_{t'}\right) = 0$$
$$\mathbb{E}\left(\bar{b}_\tau - b_t\right) = \mathbb{E}\left(\bar{b}_\tau - b_\tau - \sum_{t'=\tau+1}^{t} \lambda^{(s)}_{t'} - \tilde{\lambda}^{(s)}_{t'}\right) = \mathbb{E}\left(\bar{b}_\tau - b_\tau\right) + \sum_{t'=\tau+1}^{t} \mathbb{E}\left(\eta^{(s)}_{t'}\right) = 0$$

Assuming δ is small enough such that $\pi_t \in (\underline{\pi}, \overline{\pi})$, we define $\varepsilon_t \sim \mathcal{N}\left(0, \frac{\sigma^2}{b_t}\right)$ and apply Equation 6) along with the independence of the hidden states to get that

$$\begin{aligned}
\mathbb{E}_{\pi_\tau}\left(\pi_t\right) = \mathbb{E}_{\pi_\tau}\left(\overline{\pi} - \left(\overline{\pi} - \underline{\pi}\right)\frac{1-n_t}{b_t} + \varepsilon_t\right) &= \overline{\pi} - \left(\overline{\pi} - \underline{\pi}\right)\mathbb{E}_{\pi_\tau}\left(\frac{1-n_t}{b_t}\right) \\
&= \overline{\pi} - \left(\overline{\pi} - \underline{\pi}\right)\mathbb{E}_{\pi_\tau}\left(1-n_t\right)\mathbb{E}_{\pi_\tau}\left(\frac{1}{b_t}\right) \\
&= \overline{\pi} - \left(\overline{\pi} - \underline{\pi}\right)\left(1-\bar{n}_\tau\right)\mathbb{E}_{\pi_\tau}\left(\frac{1}{\bar{b}_\tau} - \frac{b_t - \bar{b}_\tau}{\bar{b}^2_\tau} + \frac{(b_t - \bar{b}_\tau)^2}{\bar{b}^3_\tau} - \ldots\right) \\
&\approx \overline{\pi} - \left(\overline{\pi} - \underline{\pi}\right)\frac{1-\bar{n}_\tau}{\bar{b}_\tau}\mathbb{E}_{\pi_\tau}\left(1 - \frac{b_t - \bar{b}_\tau}{\bar{b}_\tau} + \frac{(b_t - \bar{b}_\tau)^2}{\bar{b}^2_\tau}\right) \\
&= \overline{\pi} - \left(\overline{\pi} - \underline{\pi}\right)\frac{1-\bar{n}_\tau}{\bar{b}_\tau}\left(1 + \frac{\mathbb{E}_{\pi_\tau}(b_t - \bar{b}_\tau)^2}{\bar{b}^2_\tau}\right)
\end{aligned}$$

where we have taken a 2nd-order Taylor approximation of $\frac{1}{b_t}$ around $\bar{b}_\tau$. By the independence of $b_\tau, \eta^{(s)}_{\tau+1}, \ldots, \eta^{(s)}_t$ we have

$$\mathbb{E}_{\pi_\tau}\left(b_t - \bar{b}_\tau\right)^2 = \mathbb{E}_{\pi_\tau}\left(b_\tau + \sum_{t'=\tau+1}^{t} \eta^{(s)}_{t'} - \bar{b}_\tau\right)^2 = \mathbb{E}_{\pi_\tau}\left(b_t - \bar{b}_\tau\right)^2 + \sum_{t'=\tau+1}^{t} \mathbb{E}_{\pi_\tau}\left(\eta^{(s)}_{t'}\right)^2 = S^2_\tau + \frac{2\delta}{\theta^2}$$

where recall that $\delta = t - \tau$ and note that $\frac{2}{\theta^2}$ is the variance of the Laplace distribution with scale parameter $\frac{1}{\theta}$. Substituting this equation into the previous approximation yields the result.

■